BUUCTF逆向题刷题记录（一）

原创

又菜又爱倒腾 于 2021-11-02 20:55:06 发布 105 收藏 1
分类专栏： ctf_writeup 文章标签： 网络安全

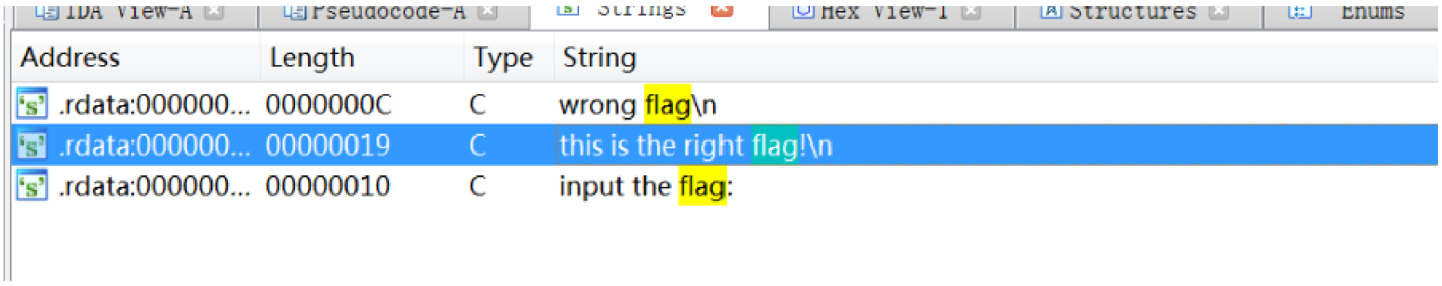
ctf_writeup 专栏收录该内容
2 篇文章 0 订阅
订阅专栏
#BUUCTF逆向题刷题记录（一）#

一、reverse1

下载reverse_2文件，拿到后丢进exeinfo PE，为64位

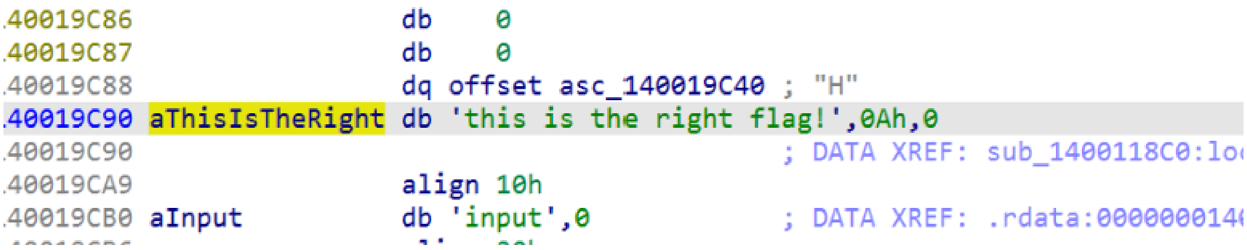
用IDA x64打开

shift+F12 查看所有字符串，搜索flag



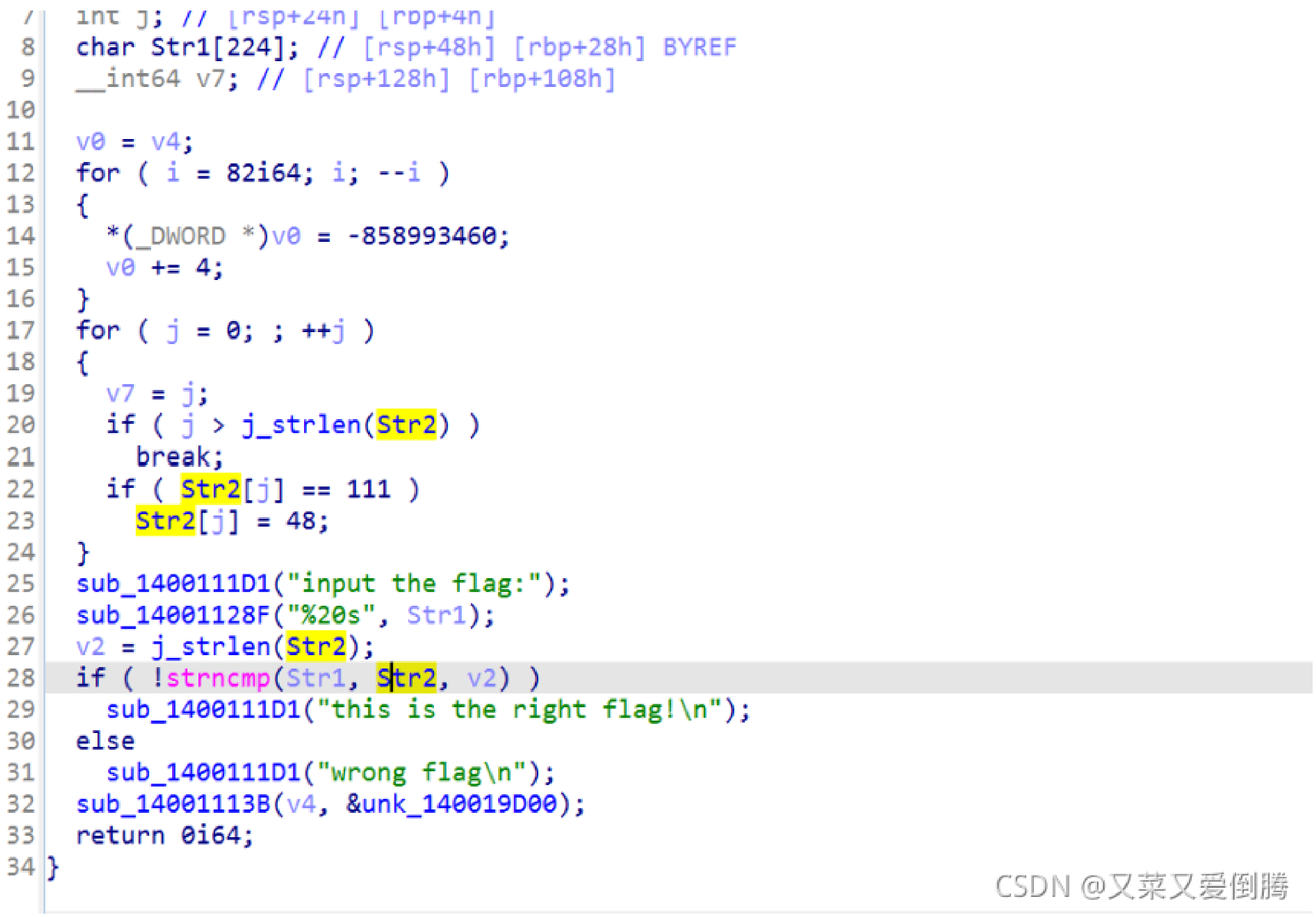
Address	Length	Type	String
.rdata:000000...	0000000C	C	wrong flag\n
.rdata:000000...	00000019	C	this is the right flag!\n
.rdata:000000...	00000010	C	input the flag:

双击进去



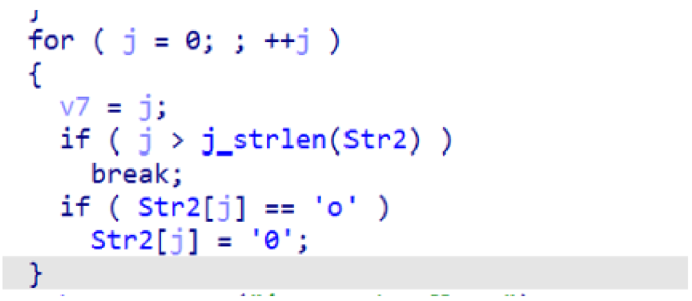
```
.40019C86                   db    0
.40019C87                   db    0
.40019C88                   dq offset asc_140019C40 ; "H"
.40019C90 aThisIsTheRight db 'this is the right flag!',0Ah,0
.40019C90                                           ; DATA XREF: sub_1400118C0:lo
.40019CA9                   align 10h
.40019CB0 aInput            db 'input',0            ; DATA XREF: .rdata:00000014
```

x查看交叉引用，跳到该处，F5查看伪代码



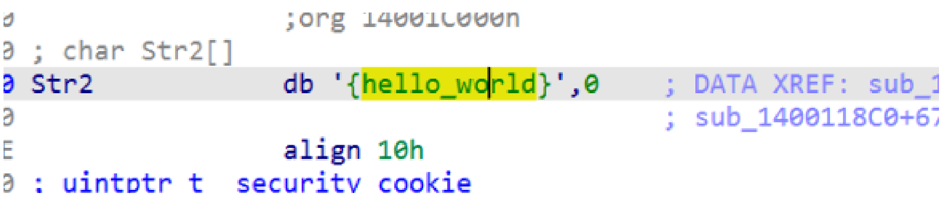
```
  int j; // [rsp+24h] [rbp+4h]
  char Str1[224]; // [rsp+48h] [rbp+28h] BYREF
  __int64 v7; // [rsp+128h] [rbp+108h]

  v0 = v4;
  for ( i = 82i64; i; --i )
  {
    *(_DWORD *)v0 = -858993460;
    v0 += 4;
  }
  for ( j = 0; ; ++j )
  {
    v7 = j;
    if ( j > j_strlen(Str2) )
      break;
    if ( Str2[j] == 111 )
      Str2[j] = 48;
  }
  sub_1400111D1("input the flag:");
  sub_14001128F("%20s", Str1);
  v2 = j_strlen(Str2);
  if ( !strncmp(Str1, Str2, v2) )
    sub_1400111D1("this is the right flag!\n");
  else
    sub_1400111D1("wrong flag\n");
  sub_14001113B(v4, &unk_140019D00);
  return 0i64;
}
```

发现存在字符串替换，r将ASCII码转换为字符。分析伪代码发现str2即为flag，但字符串str2中的’o’被替换为了’0’



```
for ( j = 0; ; ++j )
{
  v7 = j;
  if ( j > j_strlen(Str2) )
    break;
  if ( Str2[j] == 'o' )
    Str2[j] = '0';
}
```

双击跳到Str2处，将字符串替换后得到flag



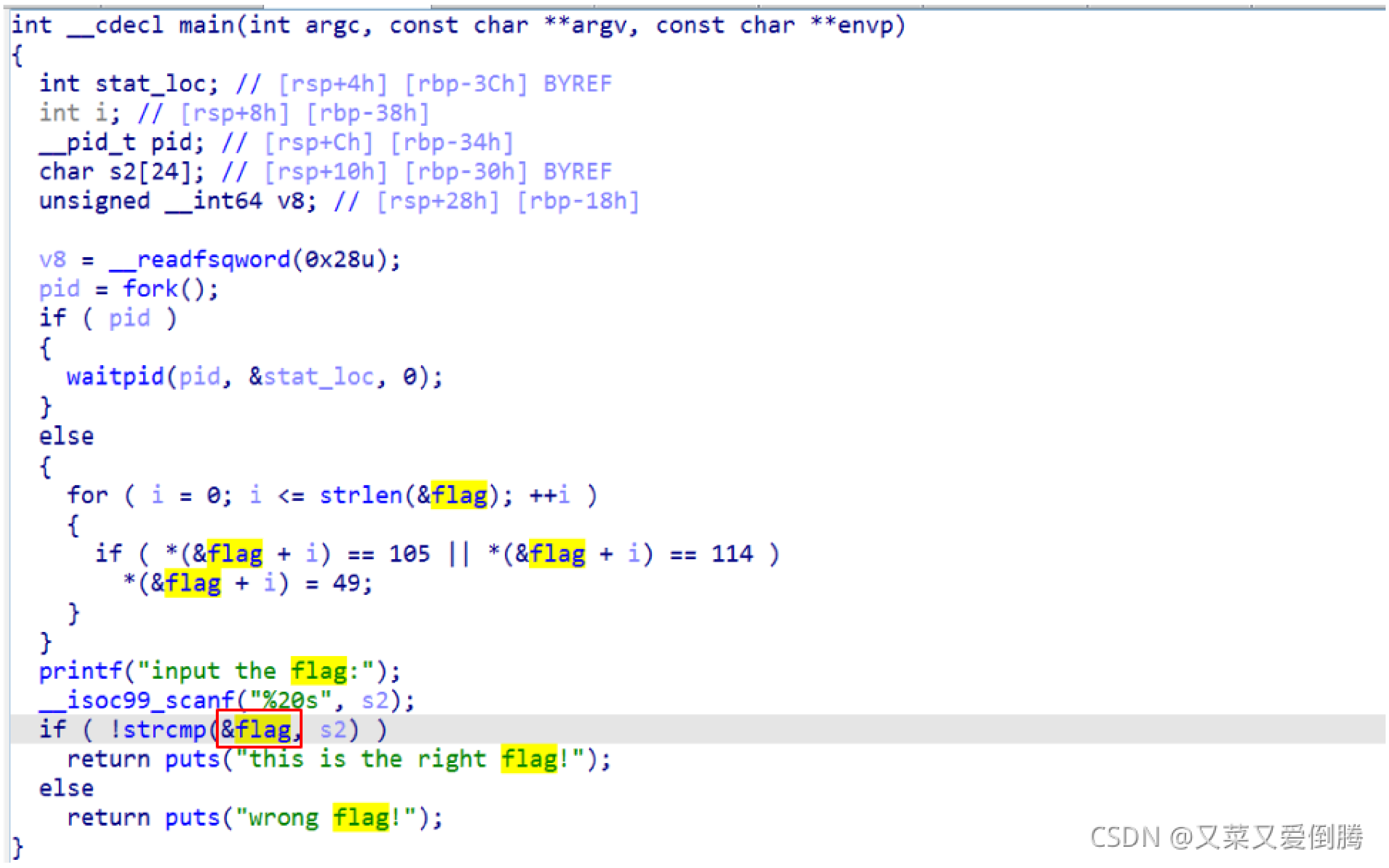
```
                ;org 14001C000h
; char Str2[]
Str2            db '{hello_world}',0     ; DATA XREF: sub_1
                                         ; sub_1400118C0+67
                align 10h
; uintptr_t security_cookie
```

二、reverse2

下载reverse_2文件，拿到后丢进exeinfo PE，为64位

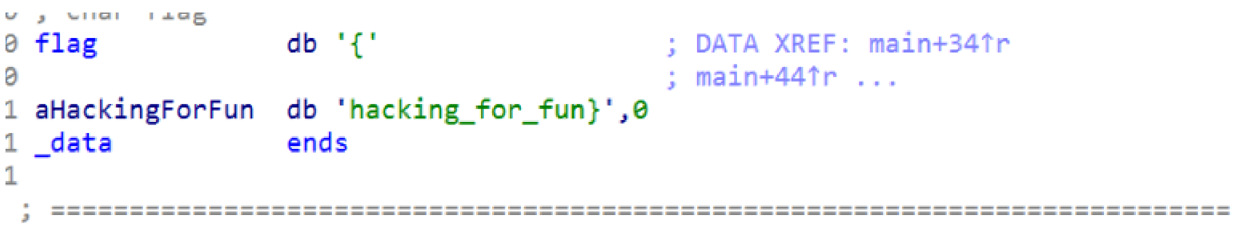
用IDA x64打开

找到main函数，F5查看伪代码



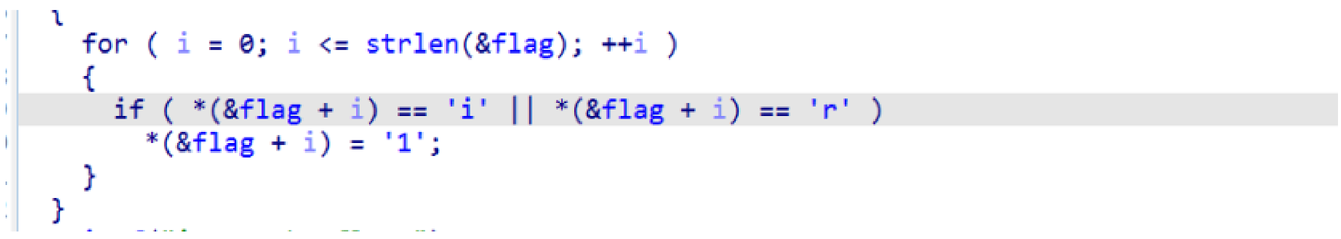
```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  int stat_loc; // [rsp+4h] [rbp-3Ch] BYREF
  int i; // [rsp+8h] [rbp-38h]
  __pid_t pid; // [rsp+Ch] [rbp-34h]
  char s2[24]; // [rsp+10h] [rbp-30h] BYREF
  unsigned __int64 v8; // [rsp+28h] [rbp-18h]

  v8 = __readfsqword(0x28u);
  pid = fork();
  if ( pid )
  {
    waitpid(pid, &stat_loc, 0);
  }
  else
  {
    for ( i = 0; i <= strlen(&flag); ++i )
    {
      if ( *(&flag + i) == 105 || *(&flag + i) == 114 )
        *(&flag + i) = 49;
    }
  }
  printf("input the flag:");
  __isoc99_scanf("%20s", s2);
  if ( !strcmp(&flag, s2) )
    return puts("this is the right flag!");
  else
    return puts("wrong flag!");
}
```

发现有一处字符串比较，双击flag点进去查看



```
0 flag            db '{'                  ; DATA XREF: main+34↑r
0                                         ; main+44↑r ...
1 aHackingForFun  db 'hacking_for_fun}',0
1 _data           ends
1
 ; ===========================================================================
```

发现flag，记下来，这还不是最后的flag，返回刚刚的伪代码，发现存在字符替换,r将ASCII码转换为字符

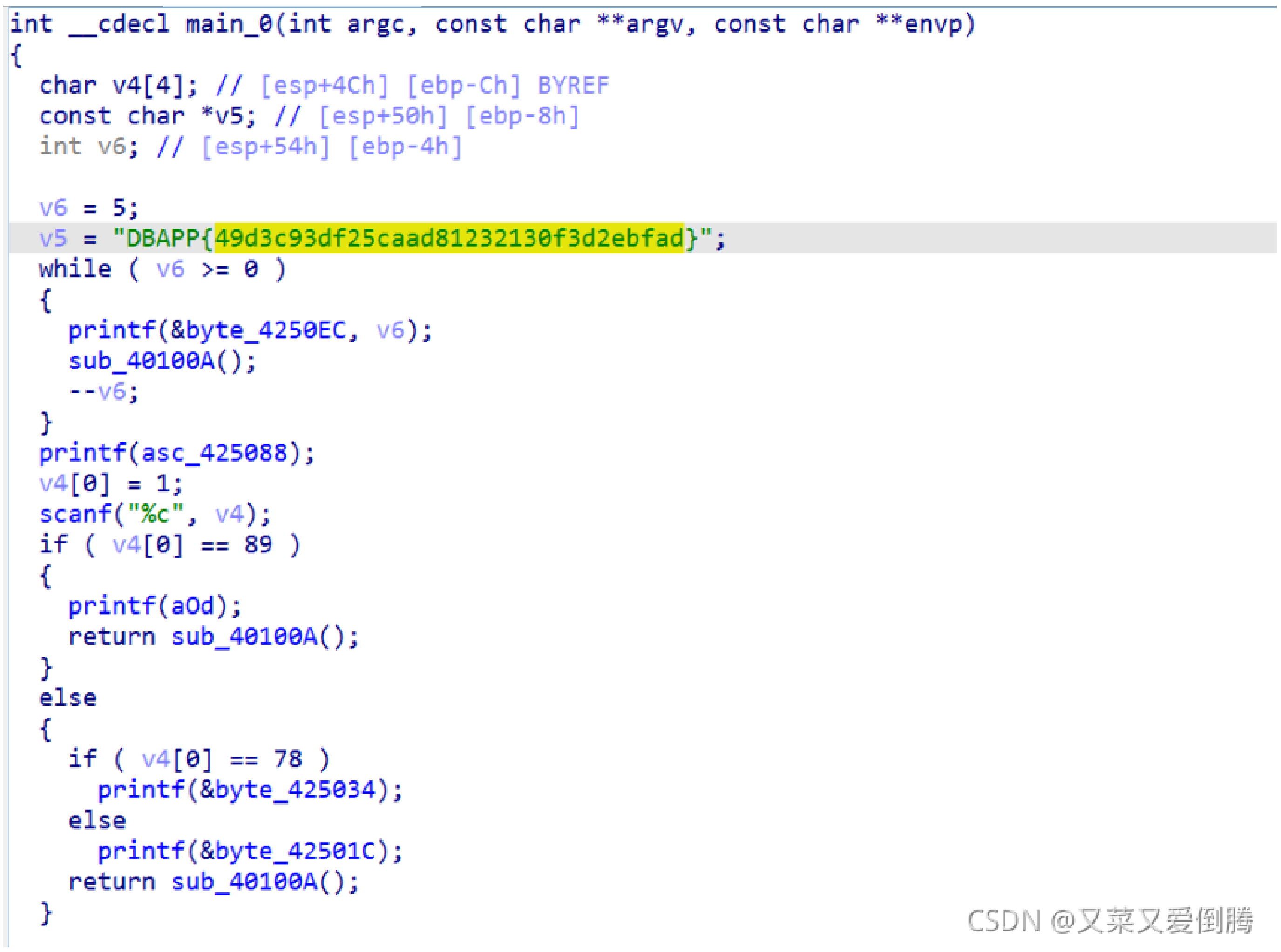


```
  {
    for ( i = 0; i <= strlen(&flag); ++i )
    {
      if ( *(&flag + i) == 'i' || *(&flag + i) == 'r' )
        *(&flag + i) = '1';
    }
  }
```

将i和r用1替换，得到flag

三、内涵的软件

下载后用IDA X86打开，找到main函数，F5查看伪代码

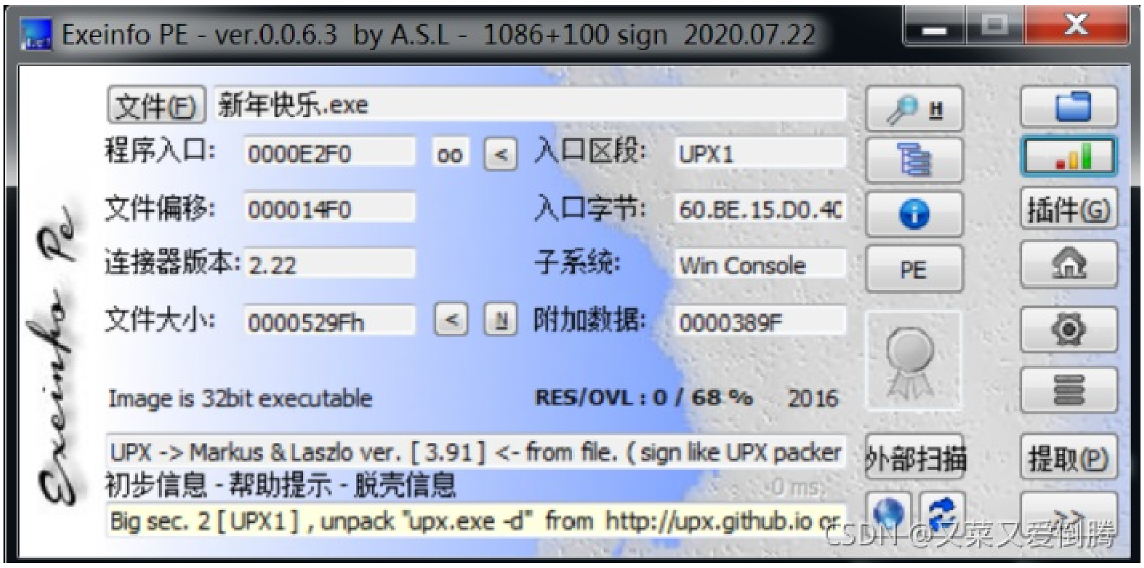


```
int __cdecl main_0(int argc, const char **argv, const char **envp)
{
  char v4[4]; // [esp+4Ch] [ebp-Ch] BYREF
  const char *v5; // [esp+50h] [ebp-8h]
  int v6; // [esp+54h] [ebp-4h]

  v6 = 5;
  v5 = "DBAPP{49d3c93df25caad81232130f3d2ebfad}";
  while ( v6 >= 0 )
  {
    printf(&byte_4250EC, v6);
    sub_40100A();
    --v6;
  }
  printf(asc_425088);
  v4[0] = 1;
  scanf("%c", v4);
  if ( v4[0] == 89 )
  {
    printf(aOd);
    return sub_40100A();
  }
  else
  {
    if ( v4[0] == 78 )
      printf(&byte_425034);
    else
      printf(&byte_42501C);
    return sub_40100A();
  }
```

找到一串可疑字符，尝试包上flag提交，成功

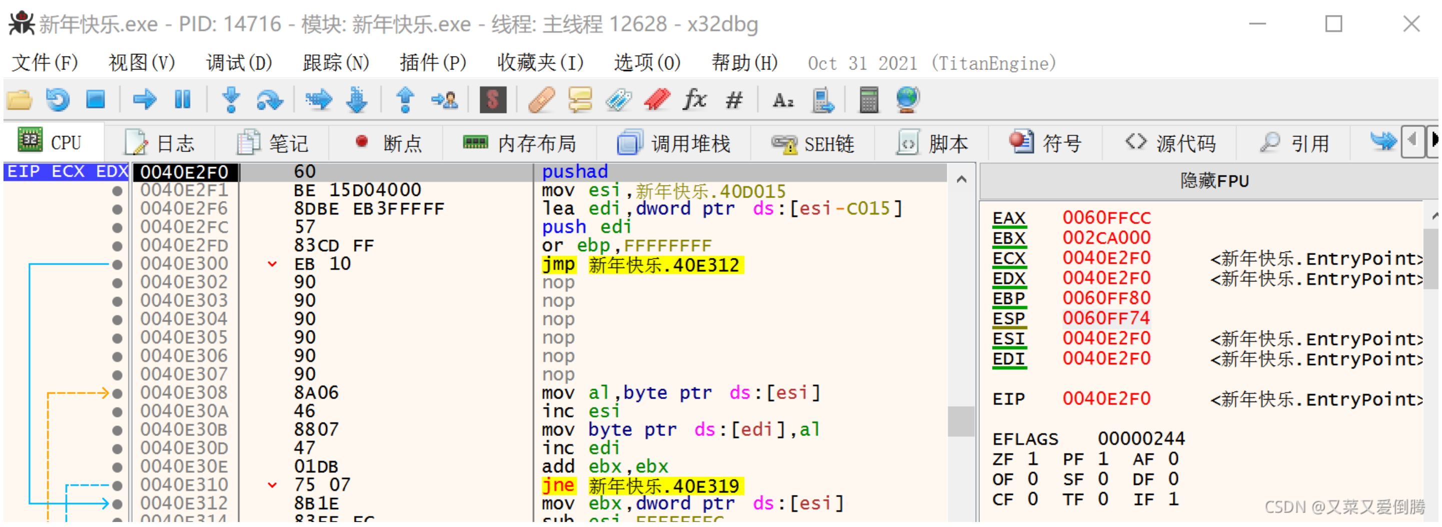
四、新年快乐

下载后丢进exeinfo PE,发现有UPX加壳



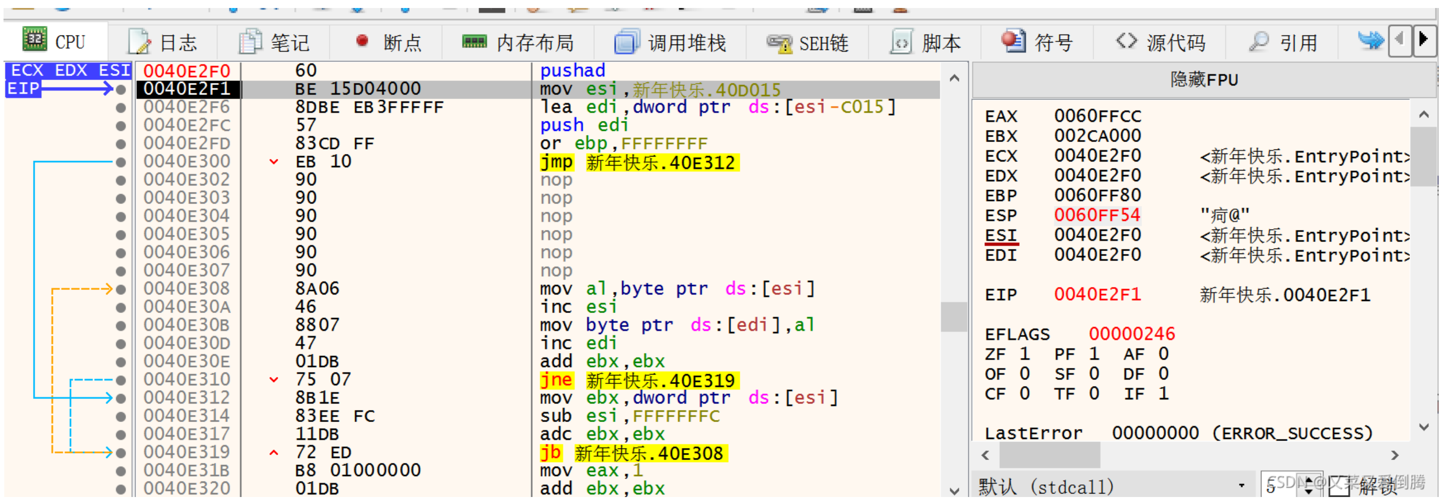
```
Exeinfo PE - ver.0.0.6.3  by A.S.L -  1086+100 sign  2020.07.22
文件(F)  新年快乐.exe
程序入口: 0000E2F0   oo  <   入口区段: UPX1
文件偏移: 000014F0           入口字节: 60.BE.15.D0.4C
连接器版本: 2.22             子系统: Win Console
文件大小: 0000529Fh  <  N   附加数据: 0000389F
Image is 32bit executable    RES/OVL : 0 / 68 %   2016
UPX -> Markus & Laszlo ver. [ 3.91 ] <- from file. ( sign like UPX packer
初步信息 - 帮助提示 - 脱壳信息                          0 ms.
Big sec. 2 [ UPX1 ] , unpack "upx.exe -d"  from  http://upx.github.io or
```

尝试手工脱壳，丢进x64DBG，F9运行两次看到Pushad



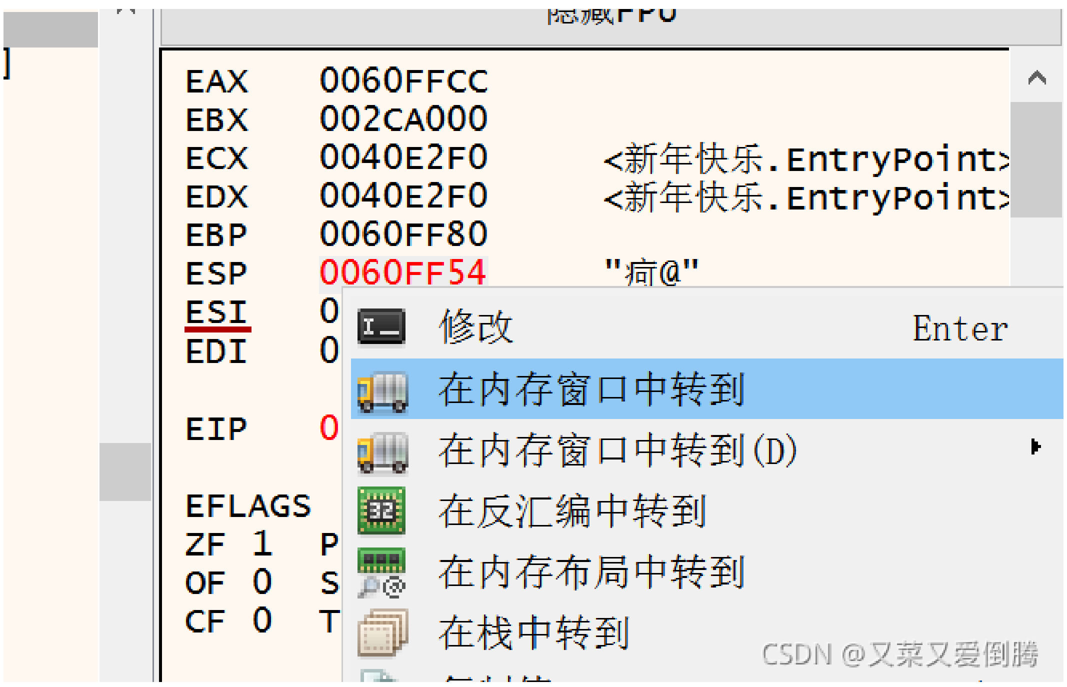
```
新年快乐.exe - PID: 14716 - 模块: 新年快乐.exe - 线程: 主线程 12628 - x32dbg
文件(F)  视图(V)  调试(D)  跟踪(N)  插件(P)  收藏夹(I)  选项(O)  帮助(H)  Oct 31 2021 (TitanEngine)
CPU  日志  笔记  断点  内存布局  调用堆栈  SEH链  脚本  符号  源代码  引用

EIP ECX EDX  0040E2F0  60              pushad
             0040E2F1  BE 15D04000     mov esi,新年快乐.40D015
             0040E2F6  8DBE EB3FFFFF   lea edi,dword ptr ds:[esi-C015]
             0040E2FC  57              push edi
             0040E2FD  83CD FF         or ebp,FFFFFFFF
             0040E300  EB 10           jmp 新年快乐.40E312
             0040E302  90              nop
             0040E303  90              nop
             0040E304  90              nop
             0040E305  90              nop
             0040E306  90              nop
             0040E307  90              nop
             0040E308  8A06            mov al,byte ptr ds:[esi]
             0040E30A  46              inc esi
             0040E30B  8807            mov byte ptr ds:[edi],al
             0040E30D  47              inc edi
             0040E30E  01DB            add ebx,ebx
             0040E310  75 07           jne 新年快乐.40E319
             0040E312  8B1E            mov ebx,dword ptr ds:[esi]

隐藏FPU
EAX   0060FFCC
EBX   002CA000
ECX   0040E2F0    <新年快乐.EntryPoint>
EDX   0040E2F0    <新年快乐.EntryPoint>
EBP   0060FF80
ESP   0060FF74
ESI   0040E2F0    <新年快乐.EntryPoint>
EDI   0040E2F0    <新年快乐.EntryPoint>

EIP   0040E2F0    <新年快乐.EntryPoint>

EFLAGS    00000244
ZF 1  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1
```

F7步进一次发现只有ESP寄存器发生变化，可根据ESP定律。



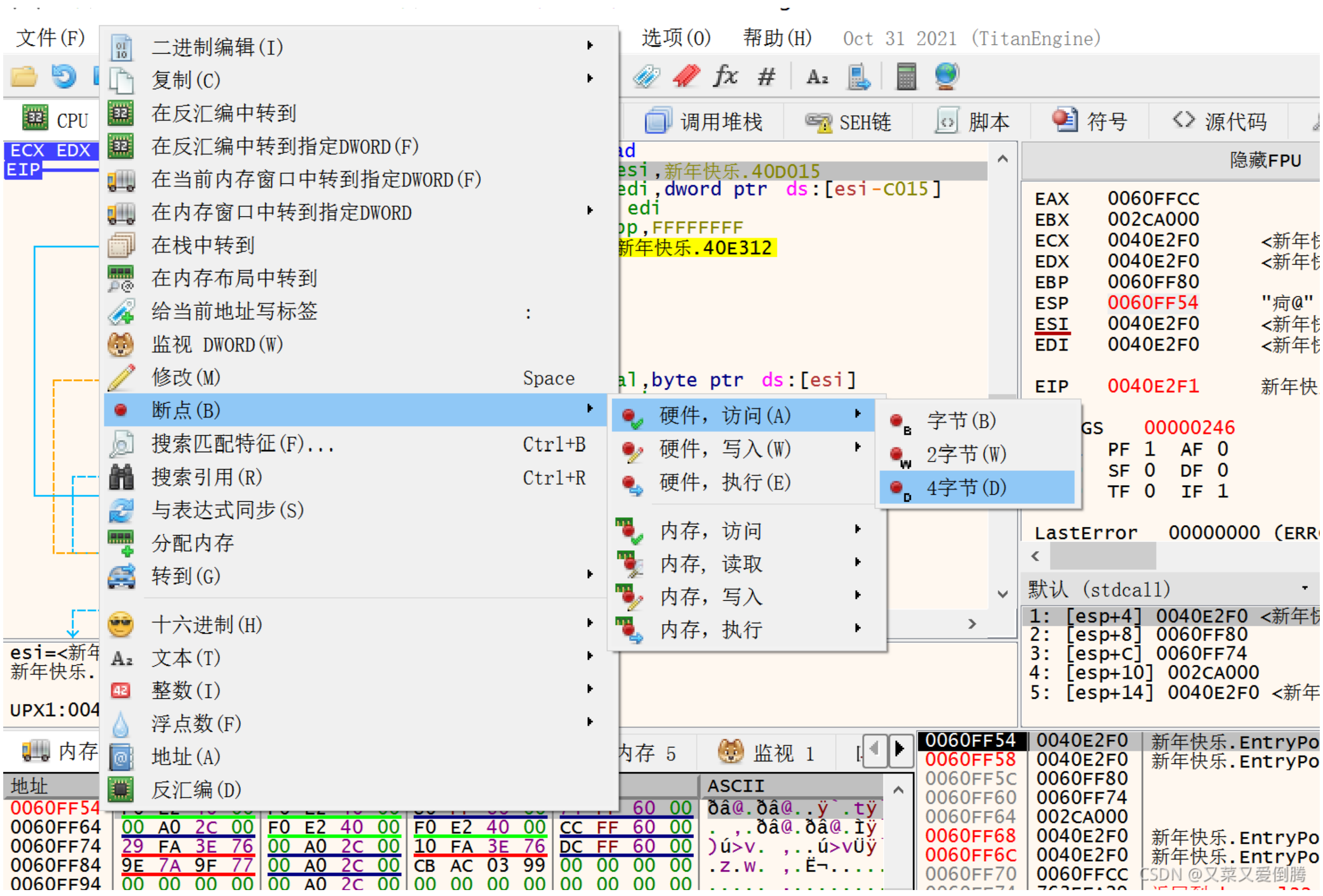
```
CPU  日志  笔记  断点  内存布局  调用堆栈  SEH链  脚本  符号  源代码  引用

ECX EDX ESI  0040E2F0  60              pushad
EIP      →   0040E2F1  BE 15D04000     mov esi,新年快乐.40D015
             0040E2F6  8DBE EB3FFFFF   lea edi,dword ptr ds:[esi-C015]
             0040E2FC  57              push edi
             0040E2FD  83CD FF         or ebp,FFFFFFFF
             0040E300  EB 10           jmp 新年快乐.40E312
             0040E302  90              nop
             0040E303  90              nop
             0040E304  90              nop
             0040E305  90              nop
             0040E306  90              nop
             0040E307  90              nop
             0040E308  8A06            mov al,byte ptr ds:[esi]
             0040E30A  46              inc esi
             0040E30B  8807            mov byte ptr ds:[edi],al
             0040E30D  47              inc edi
             0040E30E  01DB            add ebx,ebx
             0040E310  75 07           jne 新年快乐.40E319
             0040E312  8B1E            mov ebx,dword ptr ds:[esi]
             0040E314  83EE FC         sub esi,FFFFFFFC
             0040E317  11DB            adc ebx,ebx
             0040E319  72 ED           jb 新年快乐.40E308
             0040E31B  B8 01000000     mov eax,1
             0040E320  01DB            add ebx,ebx

隐藏FPU
EAX   0060FFCC
EBX   002CA000
ECX   0040E2F0    <新年快乐.EntryPoint>
EDX   0040E2F0    <新年快乐.EntryPoint>
EBP   0060FF80
ESP   0060FF54    "疴@"
ESI   0040E2F0    <新年快乐.EntryPoint>
EDI   0040E2F0    <新年快乐.EntryPoint>

EIP   0040E2F1    新年快乐.0040E2F1

EFLAGS    00000246
ZF 1  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError   00000000 (ERROR_SUCCESS)
默认 (stdcall)    5    解锁
```

选择ESP寄存器，右键在内存窗口中转到



```
EAX     0060FFCC
EBX     002CA000
ECX     0040E2F0       <新年快乐.EntryPoint>
EDX     0040E2F0       <新年快乐.EntryPoint>
EBP     0060FF80
ESP     0060FF54       "痏@"
ESI     0
EDI     0

EIP     0

EFLAGS
ZF  1   P
OF  0   S
CF  0   T
```

在内存窗口中选择地址，右键设置硬件访问断点–4字节



```
ad
esi,新年快乐.40D015
edi,dword ptr ds:[esi-C015]
 edi
bp,FFFFFFFF
新年快乐.40E312

al,byte ptr ds:[esi]
```

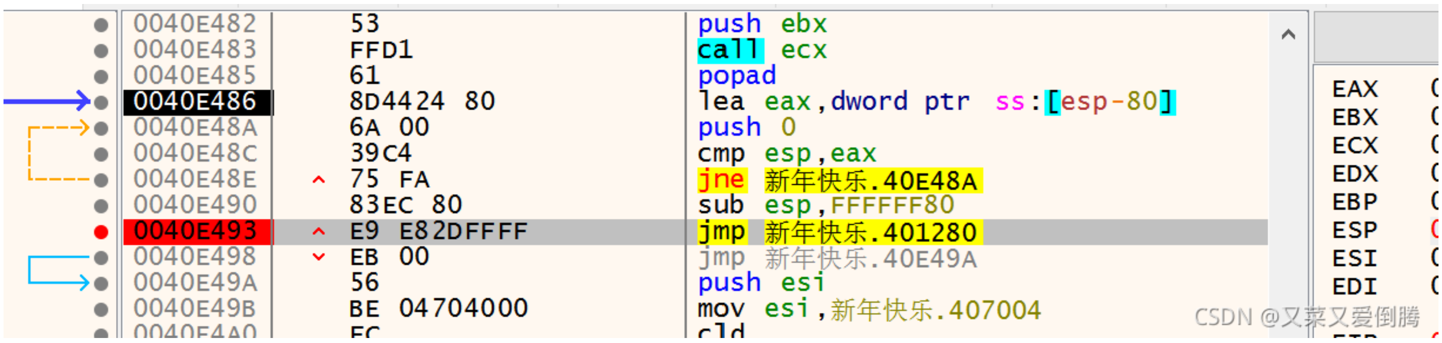
```
EAX   0060FFCC
EBX   002CA000
ECX   0040E2F0       <新年快
EDX   0040E2F0       <新年快
EBP   0060FF80
ESP   0060FF54       "痏@"
ESI   0040E2F0       <新年快
EDI   0040E2F0       <新年快

EIP   0040E2F1       新年快

GS    00000246
  PF  1   AF  0
  SF  0   DF  0
  TF  0   IF  1

LastError   00000000 (ERR

默认 (stdcall)
1: [esp+4]  0040E2F0 <新年快
2: [esp+8]  0060FF80
3: [esp+C]  0060FF74
4: [esp+10] 002CA000
5: [esp+14] 0040E2F0 <新年
```

```
esi=<新年
新年快乐.

UPX1:004
```

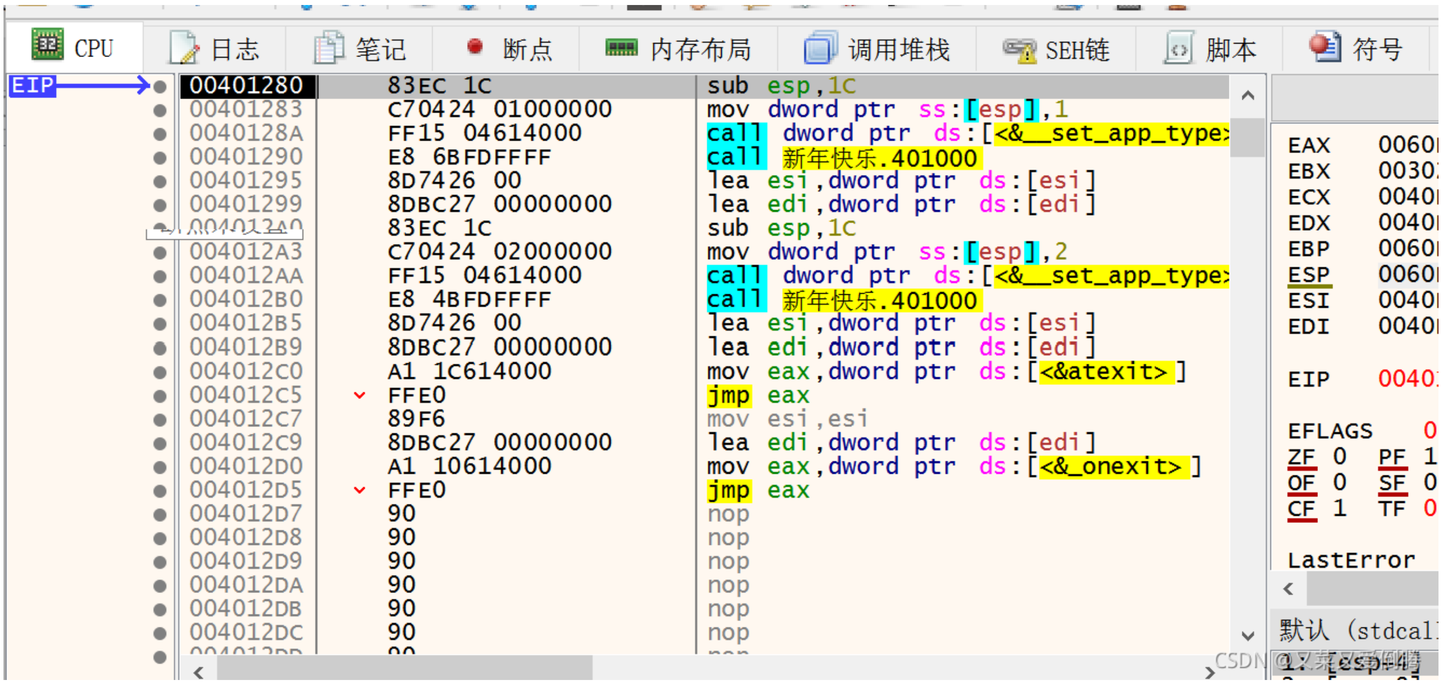
```
地址      ... ASCII
0060FF54  ...                      60 00  âà@.âà@..ÿ .tÿ
0060FF64  00 A0 2C 00  F0 E2 40 00  F0 E2 40 00  CC FF 60 00  . ,.âà@.âà@.Ìÿ
0060FF74  29 FA 3E 76  00 A0 2C 00  10 FA 3E 76  DC FF 60 00  )ú>v. ,..ú>vÜÿ
0060FF84  9E 7A 9F 77  00 A0 2C 00  CB AC 03 99  00 00 00 00  .z.w. ,.Ë¬....
0060FF94  00 00 00 00  00 A0 2C 00  00 00 00 00  00 00 00 00  ................
```

```
0060FF54  0040E2F0  新年快乐.EntryPo
0060FF58  0040E2F0  新年快乐.EntryPo
0060FF5C  0060FF80
0060FF60  0060FF74
0060FF64  002CA000
0060FF68  0040E2F0  新年快乐.EntryPo
0060FF6C  0040E2F0  新年快乐.EntryPo
0060FF70  0060FFCC
```

再次F9运行程序后，程序停下来的位置上面即popad–壳代码结束位置。在下面大跳转jmp处F2下断点。继续F9运行程序，直到jmp断点处，F7单步步进，找到OEP

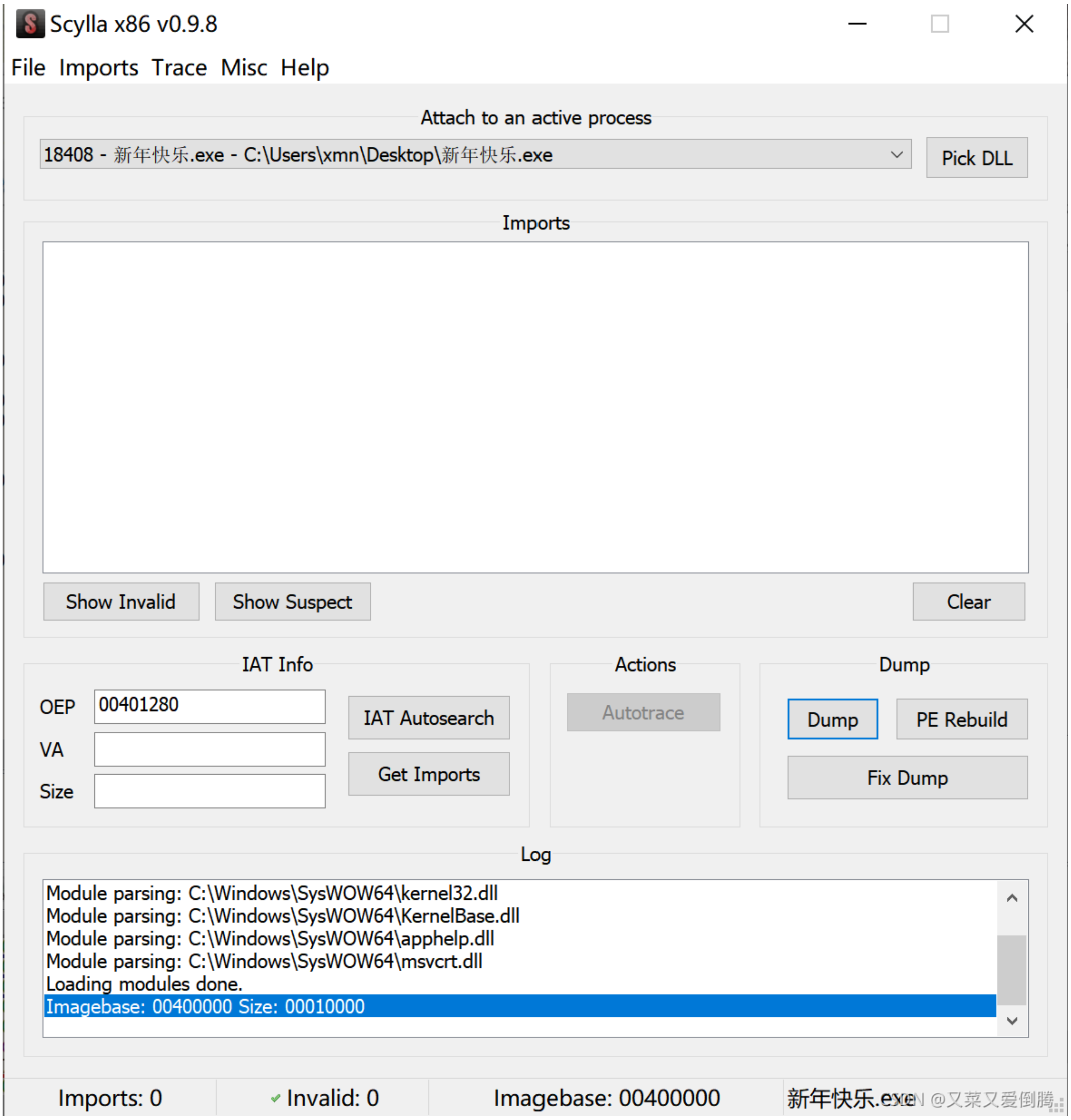


```
0040E482   53               push ebx
0040E483   FFD1             call ecx
0040E485   61               popad
0040E486   8D4424 80        lea eax,dword ptr ss:[esp-80]
0040E48A   6A 00            push 0
0040E48C   39C4             cmp esp,eax
0040E48E   75 FA            jne 新年快乐.40E48A
0040E490   83EC 80          sub esp,FFFFFF80
0040E493   E9 E82DFFFF      jmp 新年快乐.401280
0040E498   EB 00            jmp 新年快乐.40E49A
0040E49A   56               push esi
0040E49B   BE 04704000      mov esi,新年快乐.407004
0040E4A0   FC               cld
```



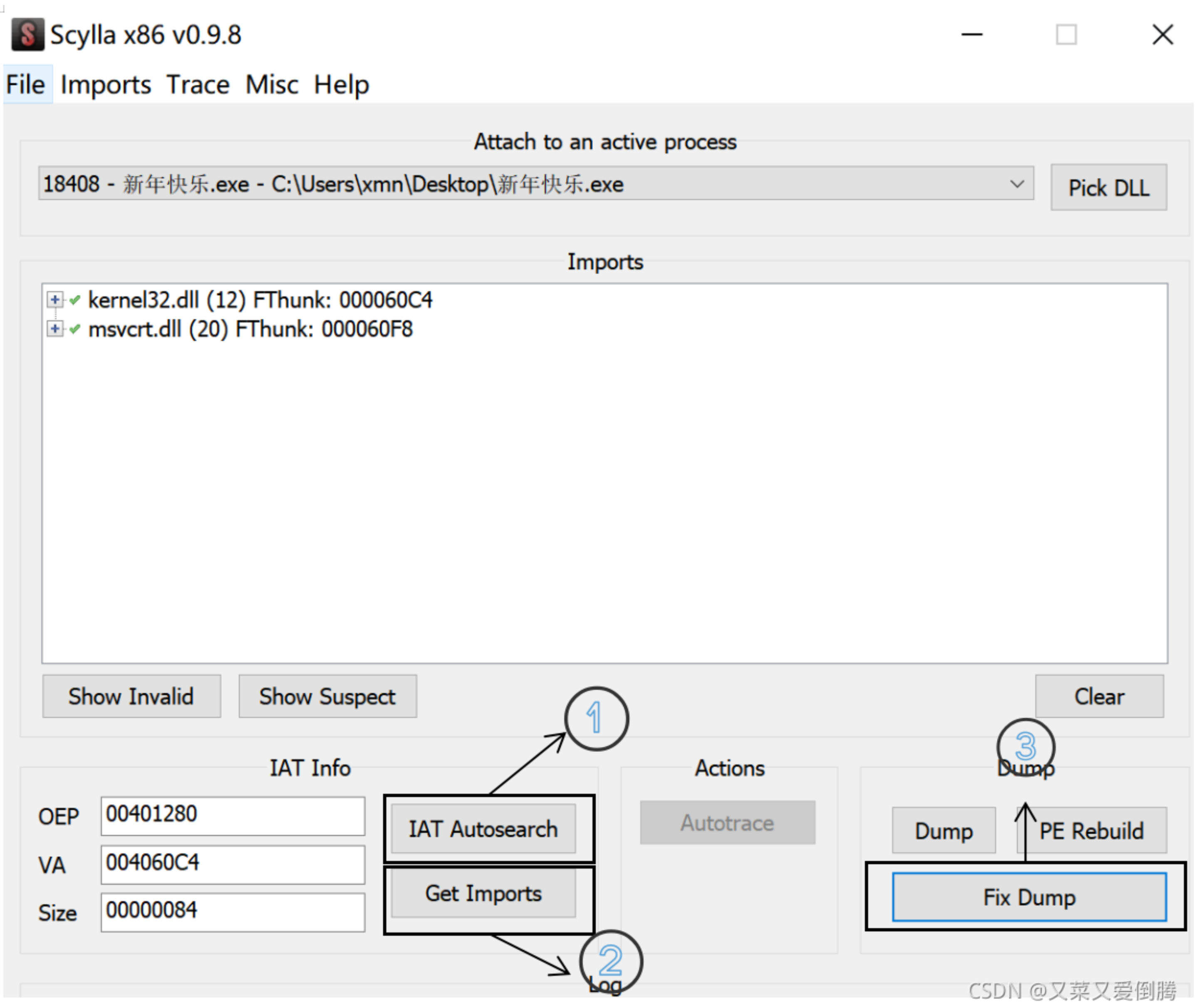
```
EIP 00401280   83EC 1C            sub esp,1C
    00401283   C70424 01000000    mov dword ptr ss:[esp],1
    0040128A   FF15 04614000      call dword ptr ds:[<&__set_app_type>
    00401290   E8 6BFDFFFF        call 新年快乐.401000
    00401295   8D7426 00          lea esi,dword ptr ds:[esi]
    00401299   8DBC27 00000000    lea edi,dword ptr ds:[edi]
    004012A0   83EC 1C            sub esp,1C
    004012A3   C70424 02000000    mov dword ptr ss:[esp],2
    004012AA   FF15 04614000      call dword ptr ds:[<&__set_app_type>
    004012B0   E8 4BFDFFFF        call 新年快乐.401000
    004012B5   8D7426 00          lea esi,dword ptr ds:[esi]
    004012B9   8DBC27 00000000    lea edi,dword ptr ds:[edi]
    004012C0   A1 1C614000        mov eax,dword ptr ds:[<&atexit>]
    004012C5   FFE0               jmp eax
    004012C7   89F6               mov esi,esi
    004012C9   8DBC27 00000000    lea edi,dword ptr ds:[edi]
    004012D0   A1 10614000        mov eax,dword ptr ds:[<&_onexit>]
    004012D5   FFE0               jmp eax
    004012D7   90                 nop
    004012D8   90                 nop
    004012D9   90                 nop
    004012DA   90                 nop
    004012DB   90                 nop
    004012DC   90                 nop
```

直接利用x64 DBG自带的Scylla将程序dump下来，核对原始地址填写正确。



```
Module parsing: C:\Windows\SysWOW64\kernel32.dll
Module parsing: C:\Windows\SysWOW64\KernelBase.dll
Module parsing: C:\Windows\SysWOW64\apphelp.dll
Module parsing: C:\Windows\SysWOW64\msvcrt.dll
Loading modules done.
Imagebase: 00400000 Size: 00010000
```

1. IAT扫描
2. 获取导入表
3. 修复Dump文件

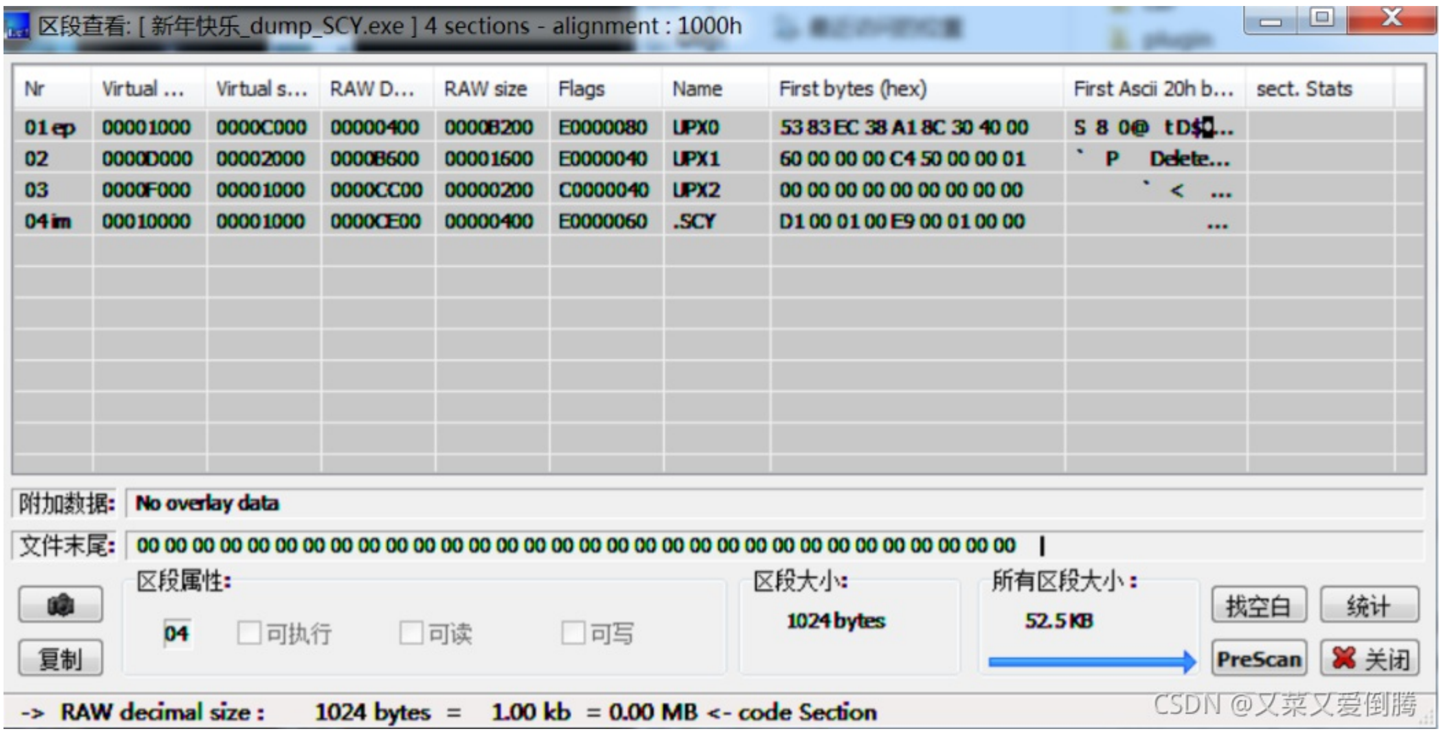


18408 - 新年快乐.exe - C:\Users\xmn\Desktop\新年快乐.exe

kernel32.dll (12) FThunk: 000060C4
msvcrt.dll (20) FThunk: 000060F8

OEP 00401280
VA 004060C4
Size 00000084

最终生成新年快乐dump.SCY.exe文件，再丢进exeinfo查看区段，脱壳成功



区段查看: [新年快乐_dump_SCY.exe] 4 sections - alignment : 1000h

Nr	Virtual ...	Virtual s...	RAW D...	RAW size	Flags	Name	First bytes (hex)	First Ascii 20h b...	sect. Stats
01 ep	00001000	0000C000	00000400	0000B200	E0000080	UPX0	53 83 EC 38 A1 8C 30 40 00	S 8 0@ tD$...	
02	0000D000	00002000	0000B600	00001600	E0000040	UPX1	60 00 00 00 C4 50 00 00 01	` P Delete...	
03	0000F000	00001000	0000CC00	00000200	C0000040	UPX2	00 00 00 00 00 00 00 00 00	` < ...	
04 im	00010000	00001000	0000CE00	00000400	E0000060	.SCY	D1 00 01 00 E9 00 01 00 00	...	

附加数据: No overlay data

文件末尾: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

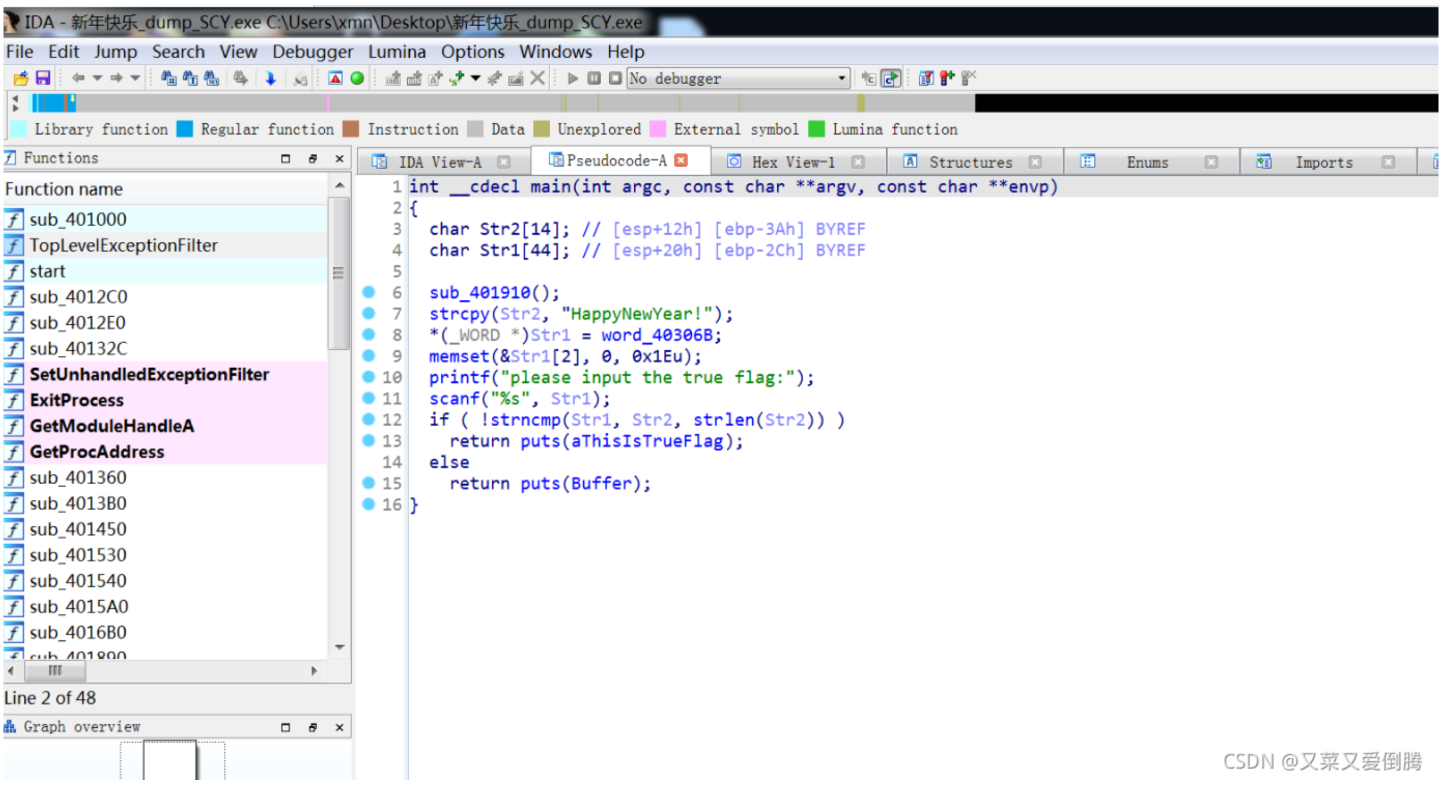
区段属性: 04 可执行 可读 可写

区段大小: 1024 bytes

所有区段大小 : 52.5 KB

-> RAW decimal size : 1024 bytes = 1.00 kb = 0.00 MB <- code Section

丢进IDA中能看到完整的函数结构，F5查看main函数伪代码



IDA - 新年快乐_dump_SCY.exe C:\Users\xmn\Desktop\新年快乐_dump_SCY.exe

Function name
sub_401000
TopLevelExceptionFilter
start
sub_4012C0
sub_4012E0
sub_40132C
SetUnhandledExceptionFilter
ExitProcess
GetModuleHandleA
GetProcAddress
sub_401360
sub_4013B0
sub_401450
sub_401530
sub_401540
sub_4015A0
sub_4016B0
sub_401890

Line 2 of 48

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  char Str2[14]; // [esp+12h] [ebp-3Ah] BYREF
  char Str1[44]; // [esp+20h] [ebp-2Ch] BYREF

  sub_401910();
  strcpy(Str2, "HappyNewYear!");
  *(_WORD *)Str1 = word_40306B;
  memset(&Str1[2], 0, 0x1Eu);
  printf("please input the true flag:");
  scanf("%s", Str1);
  if ( !strncmp(Str1, Str2, strlen(Str2)) )
    return puts(aThisIsTrueFlag);
  else
    return puts(Buffer);
}
```

分析代码，在比较函数中输入的str1会与str2进行比较，str2的"HappyNewYear!"应该就是flag的值，提交成功。

五、参考链接

使用x64dbg脱壳之开源壳upx

借助 x64dbg 的 UPX 手工脱壳

x64 DBG下载地址

六、标签

CTF逆向、X64DBG、手工脱壳